

# System Build & Load - Phase 1 Thread Assessment

April 23, 1997

Version 1.01

# System Build & Load Phase 1

## Table of Contents

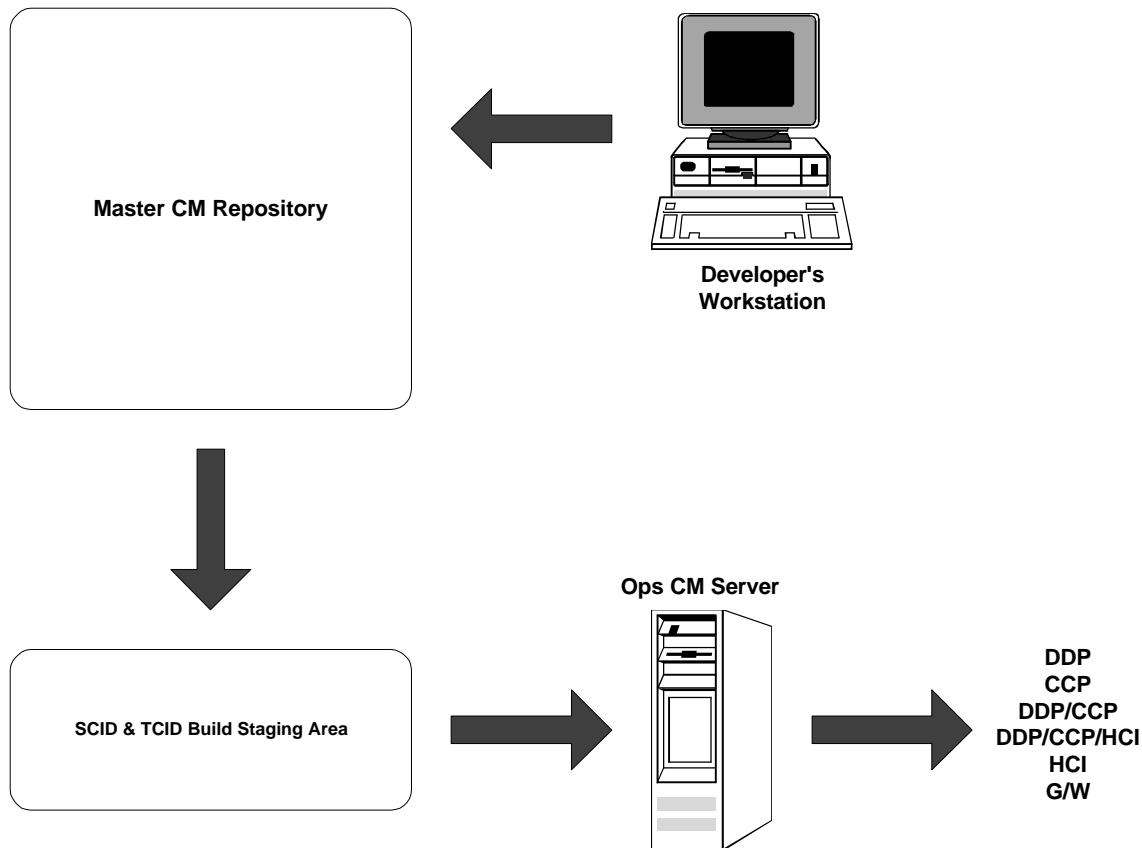
1. Introduction .....	2
1.1 System Build & Load Phase 1 Overview.....	<b>Error! Bookmark not defined.</b>
1.2 System Build & Load Phase 1 Concept.....	2
1.3 System Build & Load Phase 1 Specification.....	3
1.4 System Build & Load Phase 1 Summary.....	4
1.5 System Build & Load Phase 1 Hardware Diagram .....	5
1.6 System Build & Load Phase 1 Deliverables .....	6
1.7 System Build & Load Phase 1 Schedule.....	6
1.8 System Build & Load Phase 1 Simulation Requirements .....	6
1.9 System Build & Load Phase 1 System Test Requirements .....	6
1.10 System Build & Load Phase 1 Training Requirements .....	6
1.11 System Build & Load Phase 1 Facilities Requirements .....	7
1.12 System Build & Load Phase 1 Procurement.....	7
1.13 System Build & Load Phase 1 Dependencies .....	7
1.14 System Build & Load Phase 1 Action Items/Resolution .....	7
2. CI Assessments .....	8
2.1 System Services Assessment.....	8
2.2 System Control Assessment.....	10
2.3 CLCS Development Environment Assessment.....	12
2.4 System Build Assessment .....	13
3. COTS Products Dependencies .....	15
3.1 SW Products Dependency List.....	15

# 1. Introduction

## 1.1 System Build & Load Phase 1 Overview.

System Build & Load Phase 1 thread provides the capability to build and load OS and System Software from the Master CM repositories onto the CLCS test and operational sets. This capability is independent of Test builds yet must support multiple Test builds. This capability also provides a method for identifying repeatable System Builds called a System Configuration Identifier (SCID). The SCID, and its revision number, is a repeatable collection of products that support multiple Test builds (TCIDs) on a particular set configuration. This set configuration ranges from a set used for Application Debug, to a partially operational set which may be used for test and integration, and finally to a fully operational set.

## 1.2 System Build & Load Phase 1 Concept



### 1.3 System Build & Load Phase 1 Specification

The following requirements are derived from the SOW. The original SOW is listed below as a reference:

- This capability will build and load the OS baseline software onto the CLCS Subsystem Hardware.
- This capability will provide a means of identifying a combination of Subsystem Loads as an SCID. A Subsystem Load is comprised of all the System Software required to operate the CLCS Subsystem Hardware (i.e. DDP, CCP, DDP/CCP, DDP/CCP/HCI, HCI, Gateway).
- This capability will provide a mechanism for building an SCID (i.e. compile/link).
- This capability will provide a mechanism for loading one or more SCIDs onto the Ops CM server.
- This capability will provide a mechanism for loading an SCID onto the CLCS Subsystem Hardware from the Ops CM server.
- This capability will provide a mechanism for maintaining parallel SCIDs on an Ops CM server.
- This capability will provide a mechanism for updating an SCID on an Ops CM server.
- This capability will provide a mechanism for initializing System Software on Subsystem Hardware.

The following requirements are levied on this thread in order to complete Juno's CM shell requirements.

- This capability will provide a CM shell able to implement role based functional groups for use by the CM protection scheme, the object promotion scheme, and the build/load capabilities.
- This capability will provide a CM shell capable of supporting automated SCID/TCID builds.
- This capability will provide a CM shell for CLCS online document management.
- This capability will provide a CM shell able to implement CLCS change tracking.
- This capability will provide a CM shell that is accessible from the Office PCs and the CLCS development workstations.

#### **Original Statement of Work:**

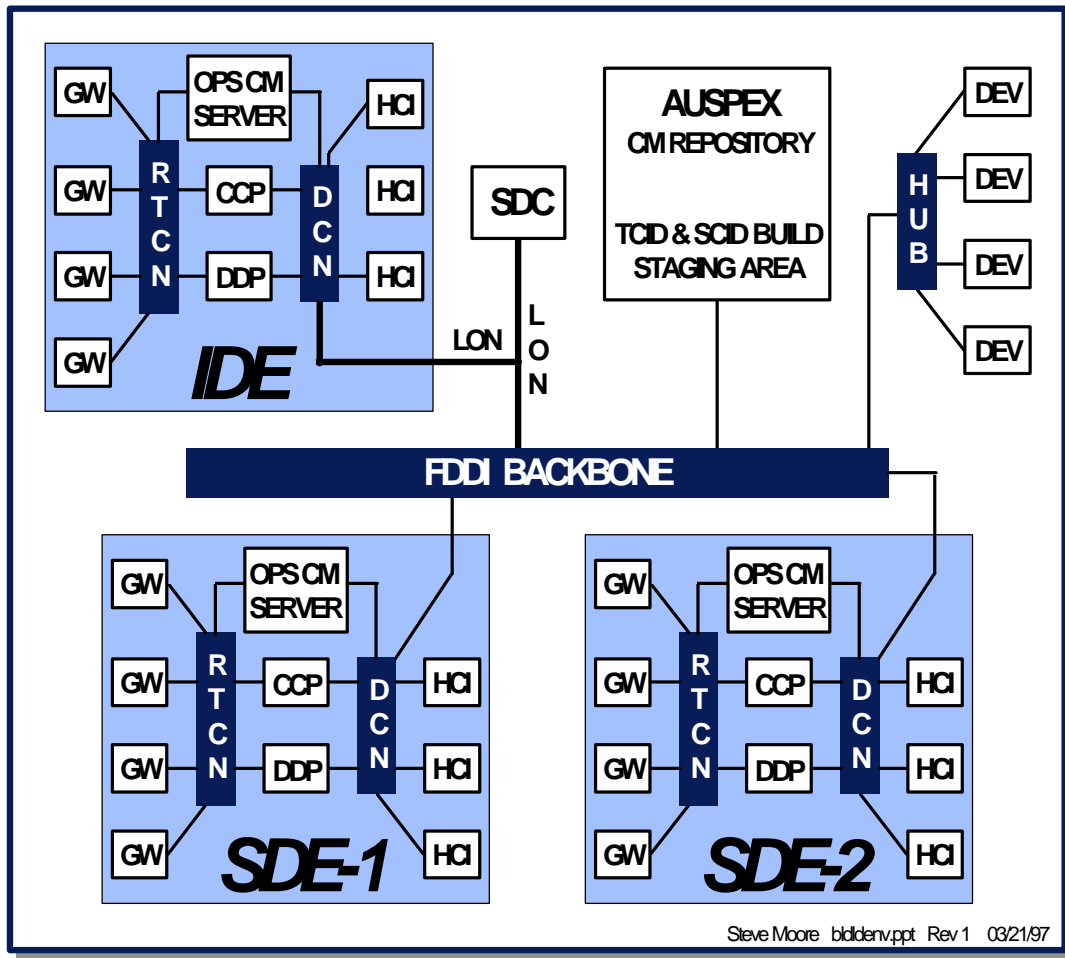
- I. Provide OS Build and Load services
- II. Integrate System Build & Load services with the CLCS Development Environment.
- III. Support definition/editing of CLCS Set Classes including:
  - A. CLCS Groups including GW's, DDP, CCP and HCI.
  - B. Logical Subsystems
    1. List of CLCS Groups in the Logical Subsystem
    2. Vendor / processor of the Logical Subsystem
- IV. Provide a System Build capability that maintains the Subsystem Builds defined in a CLCS Set Class. A System Build is identified by a SCID (System-build Configuration ID) and a revision.
- V. Provide a Build capability that creates a set of loadable directories for each subsystem.
- VI. Support parallel maintenance of multiple builds with each system build being a separate revision control thread of underlying components.
- VII. Provide an initial capability to load a System Build onto a CLCS Set.
- VIII. Provide System Build support for the following configurations:
  - A. Full Support Configuration
    1. GW's
    2. DDP and separate CCP
    3. HCI's
  - B. Limited Support Configuration
    1. GW's
    2. DDP/CCP combined
    3. HCI's
  - C. Application Debug Configuration
    1. Internal/external model support
    2. DDP/CCP/HCI combined

## 1.4 System Build & Load Phase 1 Summary

- The **CLCS Development Environment CSCI** is responsible for setting up and maintaining the CM libraries, uniquely tagging all files that are part of a build, and implementing the post-Redstone CM solution. This CSCI will set up a protection scheme that insures the integrity of the System Software source code. This CSCI will provide the users with a CVS manual and a survival guide to help ease the introduction to the Redstone CM tool. The Master CM repositories will be setup and maintained on the Auspex. This CSCI will contribute **8 Labor Months (LMs)** to support the capabilities required by this thread.
- The **System Build CSCI** is responsible for defining SCIDs, setting up and maintaining the build staging area on the Auspex, and compiling and linking all System Software, including the Gateway software. The staging area will be setup and maintained on the Auspex. This staging area will be used for both SCID and TCID files. The MCC build scripts will be evaluated for possible use by CLCS. This CSCI will contribute **15 LMs** to support the capabilities required by this thread.
- The **System Services CSCI** is responsible for defining and delivering the OS Baseline tapes as well as a set of instructions used to load these tapes onto the Boot server. For Redstone, all OS baseline software will be delivered via tape. There will be one tape per CLCS Subsystem Hardware item. This CSCI will also evaluate the MCC services AutoPilot software for use as the initialization mechanism for the CLCS system software. This CSCI will contribute **24 LMs** to support the capabilities required by this thread.
- The **System Control CSCI** is responsible for loading the SCIDs onto the CLCS Subsystem Hardware. This includes the Gateway software. This CSCI will evaluate MCC Ops CM software for use by CLCS. All loading will occur from the Ops CM server and utilize the RTCN and DCN networks to facilitate the load process. This CSCI will contribute **24 LMs** to support the capabilities required for this thread.

## 1.5 System Build & Load Phase 1 Hardware Diagram

# *Software Build/Load Environment*



## 1.6 System Build & Load Phase 1 Deliverables

- A utility that defines the makeup of an SCID.
- A utility that builds an SCID (i.e. compile/link) from the Master CM Repository.
- A utility that loads an SCID from the Build staging area onto the Ops CM server.
- A utility that loads an SCID onto the CLCS Subsystem Hardware from the Ops CM server.
- A utility that de-Configures the CLCS Subsystem Hardware (i.e. prepares the HW for a new SCID).
- A utility that initializes the System Software on the CLCS Subsystem Hardware (excepting Gateways).
- A set of OS Baseline tapes together with loading instructions. One for each CLCS Subsystem Hardware item.
- A set of initialization instructions for the Gateway software.
- Documents that describe how to use the CLCS CM tools.

## 1.7 System Build & Load Phase 1 Schedule

Please see attached schedule.

## 1.8 System Build & Load Phase 1 Simulation Requirements

There are no simulation requirements for this thread.

## 1.9 System Build & Load Phase 1 System Test Requirements

- This capability will load each OS baseline tape, using the procedures provided by the System Services CSCI, onto the Boot server and then initiate an automatic load of the CLCS Subsystem Hardware.
- This capability will test that a collection of CLCS Subsystem Loads may be defined as an SCID and modified from a user workstation.
- This capability will test that an SCID may be compiled and linked from the Master CM repositories onto the Build staging area located on the Auspex.
- This capability will test that an SCID may be transferred from the staging area, located on the Auspex, to an Ops CM server.
- This capability will test that an SCID may be loaded from an Ops CM server onto the CLCS Subsystem Hardware (including Gateway software).
- This capability will test that the system software may be initialized (including Gateway software) from the Ops CM server.
- This capability will test that an SCID may be updated once it has been loaded onto the CLCS Subsystem Hardware.

## 1.10 System Build & Load Phase 1 Training Requirements

### Training provided by this thread:

- This thread will provide training materials for the CVS/RCS CM tool.
- This thread will provide instructions on the delivery process to the Master CM repository.
- This thread will provide the users with instruction on the OS Baseline and System Software Load process (including Gateways).

### Training required by this thread:

- CM and build personnel need a few weeks to become familiar with the CVS CM tool and SDE.
- System Services CSCI needs approximately \$20K for system and network administration training from Silicon Graphics and Auspex. A detailed list of courses, schedule, and participants has been made available to Jim Roundtree and Larry Wilhelm.

## 1.11 System Build & Load Phase 1 Facilities Requirements

- Require SDE-1, SDE-2, SDE-H, and IDE configured with an Ops CM server.
- Require NFS connectivity between Master CM repository located on the Auspex, SDE-1, SDE-2, IDE, and the developer's desktops.
- Require connectivity between the CM repository and SDE-H.
- Require 9 Gigabytes of disk storage for each Ops CM server.
- Require 9 Gigabytes of disk storage for the Master CM repository and the Staging Area on the Auspex.
- Require an additional 9 Gigabytes of disk storage for the Boot server (will allow for stripping).
- Require two workstations, one for CM administration and one for System Builds. These workstations will act as compile/link engines for some of the system software (1 SGI Octane and 1 SGI O2). The Octane will need two processors and act as the CM server for all PC versions of the CM tool.
- Require access to a **Silicon Graphics Origin** and a **Silicon Graphics Indigo** for compilation of the majority of System Software.
- Require access to a **Sun development platform** for compiling Gateway software.
- Require access to the RTCN and DCN from the Ops CM server.
- Require space on the CLCS web site for training manuals and user manuals produced by this thread.
- Require a secure file cabinet for storing of OS baseline tapes and COTS products. This will also include copies of any vendor manuals. This cabinet should be fire proof to insure the integrity of the tapes. This cabinet will eventually become the CLCS tape and COTS manual library.

## 1.12 System Build & Load Phase 1 Procurement

- 4 Ops CM servers. 1 for SDE-1, 1 for SDE-2, 1 for SDE-H, and 1 for the IDE. A Silicon Graphics O2 is believed to contain sufficient horsepower to meet the requirements for Redstone.
- Four (4) 9 Gigabyte disks. One for each Ops CM server (SDE-1, SDE-2, SDE-H, and IDE).
- Four (4) 9 Gigabyte disks. One for each Boot server (SDE-1, SDE-2, SDE-H, and IDE).
- One (1) 9 Gigabyte disk for the CM repository and the Build Staging area on the Auspex.
- Two CM and Build administrator workstations (1 SGI Octane and 1 SGI O2).

## 1.13 System Build & Load Phase 1 Dependencies

- Every CLCS CSCI delivering system software needs to provide preliminary copies of their code to the Master CM repository in July. This code will be used to test the build and load software prior to the final delivery in August.
- Every CLCS CSCI delivering system software needs to provide this thread with any special build and load instructions.

## 1.14 System Build & Load Phase 1 Action Items/Resolution

- Will positional login be required for Redstone? **(TBD)**
- Should OS Baseline loads be configured for performance for Redstone? **(Yes, for DDP)**
- What type of build/load support is required for SDE-H? **(Load the Ops CM server across the net).**
- Does the CM Administration CSC need to manage CLCS Documents? Licenses? **(Yes, No)**
- Will the Redstone CSCIs provide an early drop of APIs for use by other CSCIs? **(Yes)**



## 2. CI Assessments

### 2.1 System Services Assessment

This CSCI will provide the capability to load OS baseline configurations onto the CLCS Subsystem Hardware. This CSCI will also provide the capability to initialize System Software on the CLCS Subsystem Hardware. The System Software is loaded onto the CLCS Subsystem Hardware by the System Control CSCI.

#### Function Initialization Services Work Required

1. Provide an initialization service that initializes System Software on the CLCS Subsystem Hardware.
2. Support the users in the startup and configuration of the initialization service (i.e. AutoPilot).

#### Function OS Load Services Work Required

1. Provide one OS Baseline load for each unique CLCS Subsystem Hardware item.
2. Provide procedures for installing the OS Baseline from the CM Ops server onto each CLCS Subsystem Hardware Item.

#### CSCI Assessment

CSC Name	CSCI Labor (LM)	% of CSCI	CSC LM
Initialization Service		5%	.5
OS Load Services		25%	24

#### Lines of Code

The Redstone delivery will likely be based on the re-use of the MCC JUNO ported AutoPilot Services for System Software process initialization. No changes to this software are anticipated for Redstone. The AutoPilot software will be brought to panel #3 for review.

The COTS OS utilities will be utilized for conducting the OS baseline and load process. A written set of instructions will be provided to facilitate this load process.

#### Documentation

- I. OS Baseline load instructions for each CLCS Subsystem Hardware Item.
- II. The following existing MCC documents will be provided to CLCS.

MCC AutoPilot Services API Manual Vol. I  
MCC AutoPilot Services API Manual Vol. II, Software Users Guide  
MCC AutoPilot Services Subsystem Design Specification

## Assumptions

1. AutoPilot will not be ported to the Gateway.
2. The users will utilize a different initialization approach (other than the AutoPilot services) for the Gateway startup. For Redstone, the Gateways will be manually started using a set of instructions.
3. The Network Services CSC does not have a direct dependency to this thread. It simply provides infrastructure support to the System Control CSCI.

## Open Issues

1. Will positional login be available for Redstone? This decision needs to be made early since it impacts the **OS Load Services** function?
2. Should the OS Baseline Loads be configured for performance at this stage in the project? This decision needs to be made early since it impacts the **OS Load Services** function.
3. OS related configurations, administrative and sustaining Engineering efforts are not being tracked by any threads. This includes items such as user help desk, user additions, OS configuration (i.e., disk partitioning), addition of platforms, etc.. These activities are on-going and should not be tied to a specific delivery.

## 2.2 System Control Assessment

This CSCI will provide the capability to load a System Software build or SCID onto the CLCS Subsystem Hardware. The System Software is loaded from the Ops CM server. The RTCN and DCN are used to load the software. This CSCI uses the FDDI network for Redstone to load the software from the Build Staging Area onto the Ops CM server. It will use the LON to load the Ops CM server on the IDE.

### OPS Configuration Manager Work Required

Implement the following high-level requirements:

1. Provide the capability to retrieve system software from the SDC and transfer it to the Ops CM server.
2. Provide the capability to transfer system software to Gateway, DDP, CCP, DDP/CCP, DDP/CCP/HCI, and HCI platforms.
3. Provide a GUI to perform system software transfers.
4. Provide a status of system software transfers.
5. Provide the capability to manage/correlate system software baselines and TCID baselines on a per Subsystem Hardware basis.
6. Provide the capability to initialize system software on the Subsystem Hardware via the System Services CSCI.
7. Provide a GUI to perform system software initialization.

### CSCI Assessment

CSC Name	CSCI Labor (LM)	% of CSCI	CSC LM
OPS Configuration Manager		75%	24

### Lines of Code

The Redstone delivery will be based on modifications to the MCC OPS configuration management software delivery in JUNO. The Redstone modifications will consist of 4.8KSLOC. **These costs are for both the System Build and Test Build threads.** MCC code that is re-used will be brought to panel #3 for review.

### Documentation

The following existing MCC documents will be updated and provided to CLCS:

1. OPS Configuration Management API Manual
2. OPS Configuration Management Software User Handbook
3. OPS Configuration Management As-Built Subsystem Design Specification.

### **Assumptions**

1. The current system design requires that all system and test software be loaded on the local disk of each platform. Run-time NFS mounts are not allowed.
2. NFS is allowed as a mechanism for loading System Software from the Auspex to the Ops CM server.
3. The RTCN and DNS will be used for loading all CLCS System Software.
4. The platform running the OPS configuration manager will have direct network connectivity to the Gateway, DDP, CCP, DDP/CCP, DDP/CCP/HCI, and HCI platforms requiring configuration.

### **Open Issues**

1. What requirements exist for providing OPS configuration management in the development environment?
2. What requirements exists for supporting the loading of SDE-H?

## 2.3 CLCS Development Environment Assessment

This CSCI will provide the capability to maintain and administer a Master CM Repository for all CLCS Software. The CM repository will also include a tape library used to manage and maintain all OS Baseline Loads delivered by the System Services CSCI as well as all the tapes delivered for Juno.

### Function CM Administration Work Required

1. Define, create, and maintain the Master CM Repositories on the Auspex.
2. Create helper tools for administering the Master CM repository.
3. Tag the delivered software for easy retrieval of released baselines.
4. Produce a CVS survival guide.
5. Define, create, and maintain a tape library for delivered OS baselines and delivered COTS products.
6. Produce a transition plan from the current Juno delivered CM tool to the post Redstone CM solution.
7. Implement the CLCS CM plan when it becomes available.

### CSCI Assessment

Function Name	CSCI Labor (LM)	% of CSCI	Function LM
CM Administration		30%	8

### Lines of Code

It is estimated that this CSCI will need to develop approximately 1K lines of code to maintain and administer the Master CM repository using the CVS/RCS CM tool delivered in Juno. It is also estimated that some utilities will need to be written to transfer the CVS repositories to the post Redstone CM solution. The work required to implement the post-Redstone CM solution will be presented at the Requirement's panel for this CSCI.

### Documentation

1. This CSCI will provide a CVS survival guide to help users get started with CVS.
2. This CSCI will provide a CVS manual to help users with the more detailed aspects of CVS.
3. This CSCI will provide a CM delivery guide to help users when delivering code to the Master CM repository.
4. This CSCI will provide a report evaluating COTS CM products and a recommendation for a post-Redstone CM solution. This report will be presented at the Requirements Panel.

### Assumptions

1. This CSCI assumes that the Auspex will be used as a file server for the Master CM repository.

### Open Issues

1. What type of support does this CSCI need to provide for SDE-H?

## 2.4 System Build Assessment

This CSCI will provide the capability of defining the contents of an SCID. This CSCI will also provide the capabilities of building an SCID. The build process involves the retrieval of System Software from the Master CM repositories on the Auspex, the compilation of the source files, and the linking of the load modules. The products of the build process will be placed in a staging area on the Auspex for the System Control CSCI to load onto the Ops CM server. This staging area will be under CM control.

### Function Define SCID Work Required

1. Provide a mechanism for defining the make up of an SCID.
2. Provide a mechanism for saving and retrieving a previously defined SCID.
3. Provide a mechanism for modifying the contents of an SCID. This implies that a revision process will be applied to the SCID.

### Function Manage Staging Area Work Required

1. Provide a mechanism for retrieving baseline files from the Master CM repository and transferring them to a staging area on the Auspex.
2. Provide a mechanism for determining the contents and status of the build staging area on the Auspex.
3. Provide a mechanism for cleaning up the build staging area on the Auspex.

### Function Build SCID Work Required

1. Provide a mechanism for compiling and linking the CLCS System Software.
2. Evaluate the MCC build tools for possible re-use by CLCS. Any tools used by this CSCI will be brought to panel #3 for review.

### CSCI Assessment

Function Name	CSCI Labor (LM)	% of CSCI	Function LM
Define SCID	14	25%	3
Manage Staging Area	14	15%	2
Build SCID	14	60%	10

### Lines of Code

1. The Define SCID function will require approximately 2K lines of code.
2. The Manage Staging Area function will require approximately 1K lines of code.
3. The Build SCID function will require approximately 5K lines of code.

### Documentation

1. A document that explains the build process including the management and layout of the staging area.
2. A document that explains the SCID definition process.

### **Assumptions**

1. All Redstone CSCIs will make a preliminary delivery of system software to the Master CM repository. This CSCI is dependent on this delivery in order to insure correctness of the build process for Redstone. This delivery is expected no later than Friday, July 12<sup>th</sup>.
2. All Redstone CSCIs will make their final delivery to the Master CM repository build process for Redstone no later than Monday, August 19<sup>th</sup>.

### **Open Issues**

1. Will the Redstone CSCIs provide an early drop of APIs or interfaces for use by other CSCIs? **(YES)**
2. Does the System Build CSCI need to support SDE-H? **(YES)**

### 3. COTS Products Dependencies

#### 3.1 SW Products Dependency List

At this time there is a CM/Change Tracking tool evaluation ongoing to select a post-Redstone CM solution. The following products have been reviewed:

- CCC/Harvest
- ClearCase/ClearDDTS
- Continuous/CM
- CVS/RCS w/ GNATS and other assorted freeware tools.
- Razor

The recommended CM tool for CLCS is **Razor** by Tower Concepts, Inc.